

Assignment 4: Data Wrangling

Yared S. Asfaw

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Wrangling

Directions

1. Change “Student Name” on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., “Fay_A04_DataWrangling.Rmd”) prior to submission.

The completed exercise is due on Monday, Feb 7 @ 7:00pm.

Set up your session

1. Check your working directory, load the **tidyverse** and **lubridate** packages, and upload all four raw data files associated with the EPA Air dataset. See the README file for the EPA air datasets for more information (especially if you have not worked with air quality data previously).

```
#Checking the working directory  
getwd()
```

```
## [1] "E:/EDA/Environmental_Data_Analytics_2022/Assignments"
```

```
#Loading packages  
library(dplyr)  
library(tidyverse)  
library(lubridate)
```

```
# Importing the datasets
```

```
EPAair_03_2018 <- read.csv("../Data/Raw/EPAair_03_NC2018_raw.csv", stringsAsFactors = TRUE)  
EPAair_03_2019 <- read.csv("../Data/Raw/EPAair_03_NC2019_raw.csv", stringsAsFactors = TRUE)  
EPAair_PM_2018 <- read.csv("../Data/Raw/EPAair_PM25_NC2018_raw.csv", stringsAsFactors = TRUE)  
EPAair_PM_2019 <- read.csv("../Data/Raw/EPAair_PM25_NC2019_raw.csv", stringsAsFactors = TRUE)
```

2. Explore the dimensions, column names, and structure of the datasets.

```
#The the dimension, column names, and structure of the dataset EPAair_03_2018
dim(EPAair_03_2018)
```

```
## [1] 9737 20
```

```
# The dimension of EPAair_03_2018 is 9,737 observations and 20 variables
```

```
# Column names of EPAair_03_2018
colnames(EPAair_03_2018)
```

```
## [1] "Date"
## [2] "Source"
## [3] "Site.ID"
## [4] "POC"
## [5] "Daily.Max.8.hour.Ozone.Concentration"
## [6] "UNITS"
## [7] "DAILY_AQI_VALUE"
## [8] "Site.Name"
## [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQ5_PARAMETER_CODE"
## [12] "AQ5_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
```

```
# Structure of EPAair_03_2018
str(EPAair_03_2018)
```

```
## 'data.frame': 9737 obs. of 20 variables:
## $ Date : Factor w/ 364 levels "01/01/2018","01/02/2018",...: 60 61 62
## $ Source : Factor w/ 1 level "AQ5": 1 1 1 1 1 1 1 1 1 1 ...
## $ Site.ID : int 370030005 370030005 370030005 370030005 370030005 370030005 370030005 370030005 370030005 370030005 ...
## $ POC : int 1 1 1 1 1 1 1 1 1 1 ...
## $ Daily.Max.8.hour.Ozone.Concentration: num 0.043 0.046 0.047 0.049 0.047 0.03 0.036 0.044 0.049 0.049 ...
## $ UNITS : Factor w/ 1 level "ppm": 1 1 1 1 1 1 1 1 1 1 ...
## $ DAILY_AQI_VALUE : int 40 43 44 45 44 28 33 41 45 40 ...
## $ Site.Name : Factor w/ 40 levels "", "Beaufort",...: 35 35 35 35 35 35 35 35 35 35 ...
## $ DAILY_OBS_COUNT : int 17 17 17 17 17 17 17 17 17 17 ...
## $ PERCENT_COMPLETE : num 100 100 100 100 100 100 100 100 100 100 ...
## $ AQ5_PARAMETER_CODE : int 44201 44201 44201 44201 44201 44201 44201 44201 44201 44201 ...
## $ AQ5_PARAMETER_DESC : Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 1 ...
## $ CBSA_CODE : int 25860 25860 25860 25860 25860 25860 25860 25860 25860 25860 ...
## $ CBSA_NAME : Factor w/ 17 levels "", "Asheville, NC",...: 9 9 9 9 9 9 9 9 9 9 ...
## $ STATE_CODE : int 37 37 37 37 37 37 37 37 37 37 ...
## $ STATE : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
## $ COUNTY_CODE : int 3 3 3 3 3 3 3 3 3 3 ...
```

```
## $ COUNTY : Factor w/ 32 levels "Alexander","Avery",...: 1 1 1 1 1 1 1 1
## $ SITE_LATITUDE : num 35.9 35.9 35.9 35.9 35.9 ...
## $ SITE_LONGITUDE : num -81.2 -81.2 -81.2 -81.2 -81.2 ...
```

```
# The the dimension, column names, and structure of the dataset EPAair_03_2019
dim(EPAair_03_2019)
```

```
## [1] 10592 20
```

```
# The dimension of EPAair_03_2019 is 10,592 observations and 20 variables
```

```
# Column names of EPAair_03_2019
colnames(EPAair_03_2019)
```

```
## [1] "Date"
## [2] "Source"
## [3] "Site.ID"
## [4] "POC"
## [5] "Daily.Max.8.hour.Ozone.Concentration"
## [6] "UNITS"
## [7] "DAILY_AQI_VALUE"
## [8] "Site.Name"
## [9] "DAILY_OBS_COUNT"
## [10] "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE"
## [12] "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE"
## [14] "CBSA_NAME"
## [15] "STATE_CODE"
## [16] "STATE"
## [17] "COUNTY_CODE"
## [18] "COUNTY"
## [19] "SITE_LATITUDE"
## [20] "SITE_LONGITUDE"
```

```
# Structure of EPAair_03_2019
str(EPAair_03_2019)
```

```
## 'data.frame': 10592 obs. of 20 variables:
## $ Date : Factor w/ 365 levels "01/01/2019","01/02/2019",...: 1 2 3 4 5
## $ Source : Factor w/ 2 levels "AirNow","AQS": 1 1 1 1 1 1 1 1 1 ...
## $ Site.ID : int 370030005 370030005 370030005 370030005 370030005 370030005
## $ POC : int 1 1 1 1 1 1 1 1 1 1 ...
## $ Daily.Max.8.hour.Ozone.Concentration: num 0.029 0.018 0.016 0.022 0.037 0.037 0.029 0.038 0.038
## $ UNITS : Factor w/ 1 level "ppm": 1 1 1 1 1 1 1 1 1 ...
## $ DAILY_AQI_VALUE : int 27 17 15 20 34 34 27 35 35 28 ...
## $ Site.Name : Factor w/ 38 levels "", "Beaufort",...: 33 33 33 33 33 33 33 33
## $ DAILY_OBS_COUNT : int 24 24 24 24 24 24 24 24 24 24 ...
## $ PERCENT_COMPLETE : num 100 100 100 100 100 100 100 100 100 100 ...
## $ AQS_PARAMETER_CODE : int 44201 44201 44201 44201 44201 44201 44201 44201 44201 44201
## $ AQS_PARAMETER_DESC : Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 1 ...
## $ CBSA_CODE : int 25860 25860 25860 25860 25860 25860 25860 25860 25860 25860
```

```
## $ CBSA_NAME : Factor w/ 15 levels "", "Asheville, NC", ...: 8 8 8 8 8 8 8 8 8 8 ...
## $ STATE_CODE : int 37 37 37 37 37 37 37 37 37 37 ...
## $ STATE : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
## $ COUNTY_CODE : int 3 3 3 3 3 3 3 3 3 3 ...
## $ COUNTY : Factor w/ 30 levels "Alexander", "Avery", ...: 1 1 1 1 1 1 1 1 1 1 ...
## $ SITE_LATITUDE : num 35.9 35.9 35.9 35.9 35.9 ...
## $ SITE_LONGITUDE : num -81.2 -81.2 -81.2 -81.2 -81.2 ...
```

```
# The dimension, column names, and structure of the dataset EPAair_PM_2018
dim(EPAair_PM_2018)
```

```
## [1] 8983 20
```

```
# The dimension of EPAair_PM_2018 is 8,983 observations and 20 variables
```

```
# Column names of EPAair_PM_2018
colnames(EPAair_PM_2018)
```

```
## [1] "Date" "Source"
## [3] "Site.ID" "POC"
## [5] "Daily.Mean.PM2.5.Concentration" "UNITS"
## [7] "DAILY_AQI_VALUE" "Site.Name"
## [9] "DAILY_OBS_COUNT" "PERCENT_COMPLETE"
## [11] "AQ_PARAMETER_CODE" "AQ_PARAMETER_DESC"
## [13] "CBSA_CODE" "CBSA_NAME"
## [15] "STATE_CODE" "STATE"
## [17] "COUNTY_CODE" "COUNTY"
## [19] "SITE_LATITUDE" "SITE_LONGITUDE"
```

```
# Structure of EPAair_PM_2018
str(EPAair_PM_2018)
```

```
## 'data.frame': 8983 obs. of 20 variables:
## $ Date : Factor w/ 365 levels "01/01/2018", "01/02/2018", ...: 2 5 8 11 14 17 ...
## $ Source : Factor w/ 1 level "AQ": 1 1 1 1 1 1 1 1 1 1 ...
## $ Site.ID : int 370110002 370110002 370110002 370110002 370110002 370110002 370110002 370110002 370110002 370110002 ...
## $ POC : int 1 1 1 1 1 1 1 1 1 1 ...
## $ Daily.Mean.PM2.5.Concentration: num 2.9 3.7 5.3 0.8 2.5 4.5 1.8 2.5 4.2 1.7 ...
## $ UNITS : Factor w/ 1 level "ug/m3 LC": 1 1 1 1 1 1 1 1 1 1 ...
## $ DAILY_AQI_VALUE : int 12 15 22 3 10 19 8 10 18 7 ...
## $ Site.Name : Factor w/ 25 levels "", "Blackstone", ...: 15 15 15 15 15 15 15 15 15 15 ...
## $ DAILY_OBS_COUNT : int 1 1 1 1 1 1 1 1 1 1 ...
## $ PERCENT_COMPLETE : num 100 100 100 100 100 100 100 100 100 100 ...
## $ AQ_PARAMETER_CODE : int 88502 88502 88502 88502 88502 88502 88502 88502 88502 88502 ...
## $ AQ_PARAMETER_DESC : Factor w/ 2 levels "Acceptable PM2.5 AQI & Speciation Mass", ...: 1 1 1 1 1 1 1 1 1 1 ...
## $ CBSA_CODE : int NA NA NA NA NA NA NA NA NA NA ...
## $ CBSA_NAME : Factor w/ 14 levels "", "Asheville, NC", ...: 1 1 1 1 1 1 1 1 1 1 ...
## $ STATE_CODE : int 37 37 37 37 37 37 37 37 37 37 ...
## $ STATE : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
## $ COUNTY_CODE : int 11 11 11 11 11 11 11 11 11 11 ...
## $ COUNTY : Factor w/ 21 levels "Avery", "Buncombe", ...: 1 1 1 1 1 1 1 1 1 1 ...
## $ SITE_LATITUDE : num 36 36 36 36 36 ...
## $ SITE_LONGITUDE : num -81.9 -81.9 -81.9 -81.9 -81.9 ...
```

```
# The the dimension, column names, and structure of the dataset EPAair_PM_2019
dim(EPAair_PM_2019)
```

```
## [1] 8581 20
```

```
# The dimension of EPAair_PM_2019 is 8,581 observations and 20 variables
```

```
# Column names of EPAaire_PM_2019
colnames(EPAair_PM_2019)
```

```
## [1] "Date" "Source"
## [3] "Site.ID" "POC"
## [5] "Daily.Mean.PM2.5.Concentration" "UNITS"
## [7] "DAILY_AQI_VALUE" "Site.Name"
## [9] "DAILY_OBS_COUNT" "PERCENT_COMPLETE"
## [11] "AQS_PARAMETER_CODE" "AQS_PARAMETER_DESC"
## [13] "CBSA_CODE" "CBSA_NAME"
## [15] "STATE_CODE" "STATE"
## [17] "COUNTY_CODE" "COUNTY"
## [19] "SITE_LATITUDE" "SITE_LONGITUDE"
```

```
# Structure of EPAair_PM_2019
str(EPAair_PM_2019)
```

```
## 'data.frame': 8581 obs. of 20 variables:
## $ Date : Factor w/ 365 levels "01/01/2019","01/02/2019",...: 3 6 9 12 15 18
## $ Source : Factor w/ 2 levels "AirNow","AQS": 2 2 2 2 2 2 2 2 2 2 ...
## $ Site.ID : int 370110002 370110002 370110002 370110002 370110002 370110002 ...
## $ POC : int 1 1 1 1 1 1 1 1 1 1 ...
## $ Daily.Mean.PM2.5.Concentration: num 1.6 1 1.3 6.3 2.6 1.2 1.5 1.5 3.7 1.6 ...
## $ UNITS : Factor w/ 1 level "ug/m3 LC": 1 1 1 1 1 1 1 1 1 1 ...
## $ DAILY_AQI_VALUE : int 7 4 5 26 11 5 6 6 15 7 ...
## $ Site.Name : Factor w/ 25 levels "", "Board Of Ed. Bldg.",...: 14 14 14 14 14 14 ...
## $ DAILY_OBS_COUNT : int 1 1 1 1 1 1 1 1 1 1 ...
## $ PERCENT_COMPLETE : num 100 100 100 100 100 100 100 100 100 100 ...
## $ AQS_PARAMETER_CODE : int 88502 88502 88502 88502 88502 88502 88502 88502 88502 88502 ...
## $ AQS_PARAMETER_DESC : Factor w/ 2 levels "Acceptable PM2.5 AQI & Speciation Mass",...: 1
## $ CBSA_CODE : int NA NA NA NA NA NA NA NA NA NA ...
## $ CBSA_NAME : Factor w/ 14 levels "", "Asheville, NC",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ STATE_CODE : int 37 37 37 37 37 37 37 37 37 37 ...
## $ STATE : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
## $ COUNTY_CODE : int 11 11 11 11 11 11 11 11 11 11 ...
## $ COUNTY : Factor w/ 21 levels "Avery","Buncombe",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ SITE_LATITUDE : num 36 36 36 36 36 ...
## $ SITE_LONGITUDE : num -81.9 -81.9 -81.9 -81.9 -81.9 ...
```

Wrangle individual datasets to create processed files.

3. Change date to a date object

```
# Formatting the date to a date object
EPAair_03_2018$Date <- as.Date(EPAair_03_2018$Date, format = "%m/%d/%Y")
head(EPAair_03_2018$Date)
```

```
## [1] "2018-03-01" "2018-03-02" "2018-03-03" "2018-03-04" "2018-03-05"
## [6] "2018-03-06"
```

```
EPAair_03_2019$Date <- as.Date(EPAair_03_2019$Date, format = "%m/%d/%Y")
head(EPAair_03_2019$Date)
```

```
## [1] "2019-01-01" "2019-01-02" "2019-01-03" "2019-01-04" "2019-01-05"
## [6] "2019-01-06"
```

```
EPAair_PM_2018$Date <- as.Date(EPAair_PM_2018$Date, format = "%m/%d/%Y")
head(EPAair_PM_2018$Date)
```

```
## [1] "2018-01-02" "2018-01-05" "2018-01-08" "2018-01-11" "2018-01-14"
## [6] "2018-01-17"
```

```
EPAair_PM_2019$Date <- as.Date(EPAair_PM_2019$Date, format = "%m/%d/%Y")
head(EPAair_PM_2019$Date)
```

```
## [1] "2019-01-03" "2019-01-06" "2019-01-09" "2019-01-12" "2019-01-15"
## [6] "2019-01-18"
```

4. Select the following columns: Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE

```
# Selecting the specified columns (Date, DAILY_AQI_VALUE, Site.Name,
# AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
```

```
EPAair_03_2018_sel.col <- select(EPAair_03_2018, Date, DAILY_AQI_VALUE, Site.Name,
AQS_PARAMETER_DESC, COUNTY: SITE_LONGITUDE)
head(EPAair_03_2018_sel.col)
```

```
##           Date DAILY_AQI_VALUE           Site.Name AQS_PARAMETER_DESC    COUNTY
## 1 2018-03-01             40 Taylorsville Liledoun      Ozone Alexander
## 2 2018-03-02             43 Taylorsville Liledoun      Ozone Alexander
## 3 2018-03-03             44 Taylorsville Liledoun      Ozone Alexander
## 4 2018-03-04             45 Taylorsville Liledoun      Ozone Alexander
## 5 2018-03-05             44 Taylorsville Liledoun      Ozone Alexander
## 6 2018-03-06             28 Taylorsville Liledoun      Ozone Alexander
## SITE_LATITUDE SITE_LONGITUDE
## 1      35.9138      -81.191
## 2      35.9138      -81.191
## 3      35.9138      -81.191
## 4      35.9138      -81.191
## 5      35.9138      -81.191
## 6      35.9138      -81.191
```

```
EPAair_03_2019_sel.col <- select(EPAair_03_2019, Date, DAILY_AQI_VALUE, Site.Name,
AQS_PARAMETER_DESC, COUNTY: SITE_LONGITUDE)
head(EPAair_03_2019_sel.col)
```

```
##           Date DAILY_AQI_VALUE           Site.Name AQS_PARAMETER_DESC    COUNTY
## 1 2019-01-01             27 Taylorsville Liledoun      Ozone Alexander
## 2 2019-01-02             17 Taylorsville Liledoun      Ozone Alexander
## 3 2019-01-03             15 Taylorsville Liledoun      Ozone Alexander
## 4 2019-01-04             20 Taylorsville Liledoun      Ozone Alexander
## 5 2019-01-05             34 Taylorsville Liledoun      Ozone Alexander
## 6 2019-01-06             34 Taylorsville Liledoun      Ozone Alexander
## SITE_LATITUDE SITE_LONGITUDE
## 1      35.9138      -81.191
## 2      35.9138      -81.191
## 3      35.9138      -81.191
## 4      35.9138      -81.191
## 5      35.9138      -81.191
## 6      35.9138      -81.191
```

```
EPAair_PM_2018_sel.col <- select(EPAair_PM_2018, Date, DAILY_AQI_VALUE, Site.Name,
AQS_PARAMETER_DESC, COUNTY: SITE_LONGITUDE)
head(EPAair_PM_2018_sel.col)
```

```
##           Date DAILY_AQI_VALUE           Site.Name
## 1 2018-01-02             12 Linville Falls
## 2 2018-01-05             15 Linville Falls
## 3 2018-01-08             22 Linville Falls
## 4 2018-01-11              3 Linville Falls
## 5 2018-01-14             10 Linville Falls
## 6 2018-01-17             19 Linville Falls
##           AQS_PARAMETER_DESC COUNTY SITE_LATITUDE SITE_LONGITUDE
## 1 Acceptable PM2.5 AQI & Speciation Mass Avery      35.97235      -81.93307
## 2 Acceptable PM2.5 AQI & Speciation Mass Avery      35.97235      -81.93307
## 3 Acceptable PM2.5 AQI & Speciation Mass Avery      35.97235      -81.93307
## 4 Acceptable PM2.5 AQI & Speciation Mass Avery      35.97235      -81.93307
## 5 Acceptable PM2.5 AQI & Speciation Mass Avery      35.97235      -81.93307
## 6 Acceptable PM2.5 AQI & Speciation Mass Avery      35.97235      -81.93307
```

```
EPAair_PM_2019_sel.col <- select(EPAair_PM_2019, Date, DAILY_AQI_VALUE, Site.Name,
AQS_PARAMETER_DESC, COUNTY: SITE_LONGITUDE)
head(EPAair_PM_2019_sel.col)
```

```
##           Date DAILY_AQI_VALUE           Site.Name
## 1 2019-01-03              7 Linville Falls
## 2 2019-01-06              4 Linville Falls
## 3 2019-01-09              5 Linville Falls
## 4 2019-01-12             26 Linville Falls
## 5 2019-01-15             11 Linville Falls
## 6 2019-01-18              5 Linville Falls
##           AQS_PARAMETER_DESC COUNTY SITE_LATITUDE SITE_LONGITUDE
## 1 Acceptable PM2.5 AQI & Speciation Mass Avery      35.97235      -81.93307
## 2 Acceptable PM2.5 AQI & Speciation Mass Avery      35.97235      -81.93307
```

```
## 3 Acceptable PM2.5 AQI & Speciation Mass Avery 35.97235 -81.93307
## 4 Acceptable PM2.5 AQI & Speciation Mass Avery 35.97235 -81.93307
## 5 Acceptable PM2.5 AQI & Speciation Mass Avery 35.97235 -81.93307
## 6 Acceptable PM2.5 AQI & Speciation Mass Avery 35.97235 -81.93307
```

5. For the PM2.5 datasets, fill all cells in AQS_PARAMETER_DESC with “PM2.5” (all cells in this column should be identical).

```
# Replacing the PM2.5 datasets' AQS_PARAMETER_DESC column cells with "PM2.5"
EPAair_PM_2018_sel.col$AQS_PARAMETER_DESC <- "PM2.5"
```

```
# Confirming the replacement
head(EPAair_PM_2018_sel.col$AQS_PARAMETER_DESC, 10)
```

```
## [1] "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5"
## [10] "PM2.5"
```

```
tail(EPAair_PM_2018_sel.col$AQS_PARAMETER_DESC, 10)
```

```
## [1] "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5"
## [10] "PM2.5"
```

```
EPAair_PM_2019_sel.col$AQS_PARAMETER_DESC <- "PM2.5"
```

```
# Confirming the replacement
head(EPAair_PM_2019_sel.col$AQS_PARAMETER_DESC, 10)
```

```
## [1] "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5"
## [10] "PM2.5"
```

```
tail(EPAair_PM_2019_sel.col$AQS_PARAMETER_DESC, 10)
```

```
## [1] "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5" "PM2.5"
## [10] "PM2.5"
```

6. Save all four processed datasets in the Processed folder. Use the same file names as the raw files but replace “raw” with “processed”.

```
# Saving the processed datasets
write.csv(EPAair_03_2018_sel.col, row.names = FALSE,
file = "../Data/Processed/EPAair_03_NC2018_Processed.csv")

write.csv(EPAair_03_2019_sel.col, row.names = FALSE,
file = "../Data/Processed/EPAair_03_NC2019_Processed.csv")

write.csv(EPAair_PM_2018_sel.col, row.names = FALSE,
file = "../Data/Processed/EPAair_PM25_NC2018_Processed.csv")

write.csv(EPAair_PM_2019_sel.col, row.names = FALSE,
file = "../Data/Processed/EPAair_PM25_NC2019_Processed.csv")
```


Combine datasets

- Combine the four datasets with `rbind`. Make sure your column names are identical prior to running this code.

```
# Checking and confirming the similarity of the column names of the datasets
colnames(EPAair_03_2018_sel.col) # Column names of the dataset EPAair_03_2018_sel.col
```

```
## [1] "Date"           "DAILY_AQI_VALUE"  "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"         "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```
colnames(EPAair_03_2019_sel.col) # Column names of the dataset EPAair_03_2019_sel.col
```

```
## [1] "Date"           "DAILY_AQI_VALUE"  "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"         "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```
colnames(EPAair_PM_2018_sel.col) # Column names of the dataset EPAair_PM_2018_sel.col
```

```
## [1] "Date"           "DAILY_AQI_VALUE"  "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"         "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```
colnames(EPAair_PM_2019_sel.col) # Column names of the dataset EPAair_PM_2019_sel.col
```

```
## [1] "Date"           "DAILY_AQI_VALUE"  "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"         "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```
# Combining the processed data sets using 'rbind'
EPAair_03_PM25_comb <- rbind(EPAair_03_2018_sel.col, EPAair_03_2019_sel.col,
EPAair_PM_2018_sel.col, EPAair_PM_2019_sel.col)
```

- Wrangle your new dataset with a pipe function (`%>%`) so that it fills the following conditions:
 - Filter records to include just the sites that the four data frames have in common: “Linville Falls”, “Durham Armory”, “Leggett”, “Hattie Avenue”, “Clemmons Middle”, “Mendenhall School”, “Frying Pan Mountain”, “West Johnston Co.”, “Garinger High School”, “Castle Hayne”, “Pitt Agri. Center”, “Bryson City”, “Millbrook School”. (The `intersect` function can figure out common factor levels if we didn’t give you this list...)
 - Some sites have multiple measurements per day. Use the split-apply-combine strategy to generate daily means: group by date, site, aqs parameter, and county. Take the mean of the AQI value, latitude, and longitude.
 - Add columns for “Month” and “Year” by parsing your “Date” column (hint: `lubridate` package)
 - Hint: the dimensions of this dataset should be 14,752 x 9.

```
EPAair_03_PM25_common_sites <- EPAair_03_PM25_comb%>%
  filter(Site.Name %in% c("Linville Falls", "Durham Armory", "Leggett", "Hattie Avenue",
"Clemmons Middle", "Mendenhall School", "Frying Pan Mountain", "West Johnston Co.",
"Garinger High School", "Castle Hayne", "Pitt Agri. Center", "Bryson City", "Millbrook School")) %>%
  group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%
  summarise(meanAQI = mean(DAILY_AQI_VALUE),
            meanLatitude = mean(SITE_LATITUDE),
            meanLongitude = mean(SITE_LONGITUDE))%>%
  mutate(Month = month(Date))%>%
  mutate(Year = year(Date))
```

9. Spread your datasets such that AQI values for ozone and PM2.5 are in separate columns. Each location on a specific date should now occupy only one row.

```
# Spreading the AQI values using pivot_wider
common_sites_spread <- EPAair_03_PM25_common_sites %>%
  pivot_wider(names_from = AQS_PARAMETER_DESC, values_from = meanAQI)
head(common_sites_spread)
```

```
## # A tibble: 6 x 9
## # Groups:   Date, Site.Name [6]
##   Date      Site.Name COUNTY meanLatitude meanLongitude Month   Year PM2.5 Ozone
##   <date>    <fct>    <fct>      <dbl>         <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2018-01-01 Bryson C~ Swain      35.4          -83.4     1   2018    35    NA
## 2 2018-01-01 Castle H~ New H~    34.4          -77.8     1   2018    13    NA
## 3 2018-01-01 Clemmons~ Forsy~    36.0          -80.3     1   2018    24    NA
## 4 2018-01-01 Durham A~ Durham    36.0          -78.9     1   2018    31    NA
## 5 2018-01-01 Garinger~ Meckl~    35.2          -80.8     1   2018    20    32
## 6 2018-01-01 Hattie A~ Forsy~    36.1          -80.2     1   2018    22    NA
```

10. Call up the dimensions of your new tidy dataset.

```
dim(common_sites_spread)
```

```
## [1] 8976    9
```

11. Save your processed dataset with the following file name: "EPAair_03_PM25_NC2122_Processed.csv"

```
#11 Saving the processed dataset
write.csv(common_sites_spread, row.names = FALSE,
file = "../Data/Processed/EPAair_03_PM25_NC2122_Processed.csv")
```

Generate summary tables

- 12a. Use the split-apply-combine strategy to generate a summary data frame from your results from Step 9 above. Data should be grouped by site, month, and year. Generate the mean AQI values for ozone and PM2.5 for each group.

```
summary_table_common_sites <- common_sites_spread %>%
  group_by(Site.Name, Month, Year) %>%
  summarise(PM2.5mean = mean(PM2.5),
            Ozonemean = mean(Ozone))
head(summary_table_common_sites)
```

```
## # A tibble: 6 x 5
## # Groups:   Site.Name, Month [3]
##   Site.Name    Month  Year PM2.5mean Ozonemean
##   <fct>        <dbl> <dbl>     <dbl>     <dbl>
## 1 Bryson City      1  2018      38.9       NA
## 2 Bryson City      1  2019      29.8       NA
## 3 Bryson City      2  2018      27.2       NA
## 4 Bryson City      2  2019      33.0       NA
## 5 Bryson City      3  2018      34.7      41.6
## 6 Bryson City      3  2019       NA      42.5
```

12b. BONUS: Add a piped statement to 12a that removes rows where both mean ozone and mean PM2.5 have missing values.

```
# Forming a summary table with missing values of mean Ozone and mean #PM2.5 removed
summary_table1 <- common_sites_spread %>%
  group_by(Site.Name, Month, Year) %>%
  summarise(PM2.5mean = mean(PM2.5),
            Ozonemean = mean(Ozone)) %>%
  drop_na(PM2.5mean, Ozonemean)

# Checking na.omit function to form the same summary table as above
summary_table2 <- common_sites_spread %>%
  group_by(Site.Name, Month, Year) %>%
  summarise(PM2.5mean = mean(PM2.5),
            Ozonemean = mean(Ozone)) %>%
  na.omit(PM2.5mean, Ozonemean)
```

```
# Looking at the columns of PM2.5mean and Ozonemean
head(summary_table1) #summary table formed using the drop.na function
```

```
## # A tibble: 6 x 5
## # Groups:   Site.Name, Month [5]
##   Site.Name    Month  Year PM2.5mean Ozonemean
##   <fct>        <dbl> <dbl>     <dbl>     <dbl>
## 1 Bryson City      3  2018      34.7      41.6
## 2 Bryson City      4  2018      28.2      44.5
## 3 Bryson City      4  2019      26.7      45.4
## 4 Bryson City      7  2019      33.6      30.4
## 5 Bryson City      9  2018      25.1      25.4
## 6 Bryson City     10  2018      31.3       31
```

```
head(summary_table2) #summary table formed using the na.omit function
```

```
## # A tibble: 6 x 5
```

```
## # Groups:   Site.Name, Month [5]
##   Site.Name  Month  Year PM2.5mean Ozone
##   <fct>      <dbl> <dbl>    <dbl>    <dbl>
## 1 Bryson City    3  2018     34.7     41.6
## 2 Bryson City    4  2018     28.2     44.5
## 3 Bryson City    4  2019     26.7     45.4
## 4 Bryson City    7  2019     33.6     30.4
## 5 Bryson City    9  2018     25.1     25.4
## 6 Bryson City   10  2018     31.3      31
```

```
# No NA value in both PM2.5mean and Ozone columns
```

13. Call up the dimensions of the summary dataset.

```
# The dimensions of the summary dataset before and after removing the missing
# values of mean Ozone and mean #PM2.5
dim(summary_table_common_sites) # The dimension before removing the missing values
```

```
## [1] 308  5
```

```
# The dimension of the summary dataset before removing the missing values of
# mean Ozone and mean PM2.5 is 308 observations and 5 variables
```

```
dim(summary_table1) # The dimension after removing the missing values
```

```
## [1] 101  5
```

```
# The dimension of the summary dataset after removing the missing values of
# mean Ozone and mean PM2.5 is 101 observations and 5 variables
```

14. Why did we use the function `drop_na` rather than `na.omit`?

Answer: I don't see any difference in using both functions. When running both result in the same output.